NAT'L INST. OF STAND & TECH R.I.C.
A11104 062711

NATIONAL INSTITUTE OF STANDARDS &
TECHNOLOGY
Research Information Center
Gaithersburg, MD 20899

A11103 437599

NIST
PUBLICATIONS

United States Department of Commerce
National Institute of Standards and Technology

NISTIR 89-3916

TIME DOMAIN FREQUENCY STABILITY CALCULATED FROM THE FREQUENCY DOMAIN DESCRIPTION:

Use of the SIGINT Software Package to Calculate Time Domain Frequency Stability From the Frequency Domain

F. L. Walls John Gary Abbie O'Gallagher
Roland Sweet Linda Sweet

National Institute of Standards and Technology
U.S. Department of Commerce
Boulder, Colorado 80303-3328

September 1989

QC
100
.U56
89-3916
1989
C.2

NATIONAL INSTITUTE OF STANDARDS &
TECHNOLOGY
Research Information Center
Gaithersburg, MD 20899

DATE DUE

Demco, Inc. 38-293

NIST
QC100
.U56
89-3916
1989
C.2

NISTIR 89-3916

TIME DOMAIN FREQUENCY STABILITY CALCULATED FROM THE FREQUENCY DOMAIN DESCRIPTION:

Use of the SIGINT Software Package to Calculate Time Domain Frequency Stability From the Frequency Domain

F. L. Walls John Gary Abbie O'Gallagher
Roland Sweet Linda Sweet

National Institute of Standards and Technology
U.S. Department of Commerce
Boulder, Colorado 80303-3328

September 1989

U.S. DEPARTMENT OF COMMERCE, Robert A. Mosbacher, Secretary

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, Raymond G. Kammer, Acting Director

Contents

Abstract vii

1 Introduction 1

2 Running the Program 3

2.1 Input Modes 3

2.1.1 The "prompting" input mode 3

2.1.2 The "direct" input mode 3

2.2 Output Modes 4

2.2.1 Printing 4

2.2.2 Plotting 4

2.2.3 Quitting and Saving Results 5

2.3 Input Parameters 5

2.4 Internal Constants 6

3 The Numerical Method 7

4 Portability 10

5 References 12

Appendix 22

List of Figures

1 $\sigma_y(\tau)$ versus τ for the five common power-law noise types in the limit that $2\pi f_h$ is large compared to 1 and an infinitely sharp filter is used. Curve a is for random-walk frequency modulation, $S_y(f) = h_{-2}f^{-2}$. Curve b is for flicker frequency modulation, $S_y(f) = h_{-1}f^{-1}$. Curve c is for white frequency modulation, $S_y(f) = h_0$. Curve d is for flicker phase modulation, $S_y(f) = h_1 f$. Curve e is for white phase modulation, $S_y(f) = h_2 f^2$. 13

2 $\sigma_y(\tau)$ for white phase modulation ($\alpha = 2$) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b, and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively. 14

3 $\sigma_y(\tau)$ for flicker phase frequency modulation ($\alpha = 1$) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively. 15

4 $\sigma_y(\tau)$ for white frequency modulation ($\alpha = 0$) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively. 16

5 $\sigma_y(\tau)$ for flicker frequency modulation ($\alpha = -1$) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively. 17

6 $\sigma_y(\tau)$ for random-walk frequency modulation ($\alpha = -2$) as a function of measurement time and measurement bandwidth, f_h for π. Curves a, b and c have an infinitely sharp filter with width, $f_h = 16$ Hz, $f_h = 0.016$ Hz, $f_h = 0.0016$ Hz respectively. Curves d and e have a single pole filter width, $f_h = 0.016$ Hz and $f_h = 0.0016$ respectively. 18

7 Ratio of $\sigma_y^2(\tau)$ to $mod\sigma_y^2(\tau)$ as a function of n, the number of points averaged to obtain $mod\sigma_y(\tau)$. The measurement time $\tau = n\tau_0$, where τ_0 is the minimum data interval. 19

List of Tables

I Ratio of $mod\sigma_y^2(\tau)$ to $\sigma_y^2(\tau)$ versus n, for common power-law noise types $S_y(f) = h_\alpha f^\alpha$. n is the number of time or phase samples averaged to obtain $mod\sigma_y^2(\tau = n\tau_0)$ where τ_0 is the minimum sample time, and ω_h is 2π times the measurement bandwidth f_h. 20

II Asymptotic forms of $\sigma_y^2(\tau)$ for various power-law noise types and two filter types. Note: $\omega_h/2\pi = f_h$ is the measurement system bandwidth – often called the high-frequency cutoff. ln $\equiv log_e$. 21

TIME DOMAIN FREQUENCY STABILITY CALCULATED FROM THE FREQUENCY DOMAIN DESCRIPTION:

Use of the SIGINT Software Package to Calculate Time Domain Frequency Stability From the Frequency Domain

F. L. Walls John Gary Abbie O'Gallagher
Roland Sweet Linda Sweet

National Institute of Standards and Technology
Boulder, Colorado 80303-3328

We describe the use of SIGINT, a new interactive software package developed by the National Institute of Standards and Technology, which facilitates the calculation of time domain frequency stability in terms of the Allan variance, $\sigma_y^2(\tau)$, or mod$\sigma_y^2(\tau)$ as a function of measuring time from frequency domain data. Except for the graphic output, the code is written in standard FORTRAN 77 and runs on AT compatible computers that have a math co-processor. It also runs on many other systems; however, calls to an available graphics library will need to be substituted for those that are included in this version. The program uses either a user defined function for the input noise or default functions that describe the noise types commonly found in oscillators, amplifiers, frequency multipliers, frequency dividers, and general signal processing equipment. These default functions make it simple to analyze the time domain frequency stability as a function of measuring bandwidth using realistic first-, second-, or third- order low-pass filters or the simplified infinitely sharp cutoff parameter f_h. The default functions are also set up to examine the effect of various servo parameters on the performance of a frequency source locked to a frequency reference.

Key words: Allan variance; frequency lock loop analysis; modified Allan variance; phase lock loop analysis; spectral density of frequency stability; time domain frequency stability.

1 Introduction

SIGINT is an interactive software package designed to facilitate the calculation of time domain frequency stability from data in the frequency domain. Both the Allan variance (two-sample variance), $\sigma_y^2(\tau)$, and the modified Allan variance, $\mathrm{mod}\sigma_y^2(\tau)$, can be calculated as functions of measurement time. Except for the graphic output, the code is written in standard FORTRAN 77 to run on AT and compatibles that have a math co-processor (see §4 for details on portability). The actual output of the software is the square root of the variance. The equations are [1,2,3,4]

$$\sigma_y(n\tau_0) = \left[2\int_0^{f_h} S_y(f)\frac{\sin^4(\pi f n\tau_0)}{(\pi f n\tau_0)^2}df\right]^{\frac{1}{2}}, and$$

$$\mathrm{mod}\sigma_y(n\tau_0) = \left[\frac{2}{n^4\pi^2\tau_0^2}\left\{n\int_0^{f_h}\frac{S_y(f)}{f^2}\sin^4(\pi f n\tau_0)\,df + 2\int_0^{f_h}\sum_{k=1}^{n-1}(n-k)\frac{S_y(f)}{f^2}\cos(2\pi f k\tau_0)\sin^4(\pi f n\tau_0)\,df\right\}\right]^{\frac{1}{2}}$$

where the measurement time is given by $n\tau_0$ with τ_0 being the minimum measurement time.

The input parameters are expressed in terms of the spectral density of frequency fluctuations, $S_y(f)$. Conversion from phase noise to $S_y(f)$ is simply

$$S_y(f) = \frac{f^2}{\nu^2}S_\phi(f), and$$
$$S_\phi(f) = \frac{\nu^2}{f^2}S_y(f).$$

You may choose to supply your own function $S_y(f)$, or to use the functions which are built into the package. Use the parameter SELSY (see below) to communicate this choice to the package. If you use your own function, you must modify the empty FORTRAN function, SYF(f), within the package. The built in function has the form:

$$S_y(f) = \frac{C_1 f^{-2} + C_2 f^{-1} + C_3 + C_4 f + C_5 f^2}{K(f)M(f)} + C_6\ .$$

The values of the C_i are stored in array C. The filter function $K(f)$ must be chosen from the following four functions by setting the value of SELK (SELK=0, 1, 2 and 3 respectively).

$$
\begin{aligned}
K(f) &= 1, \\
K(f) &= \left(1+\frac{1}{K_1 f}\right)^2, \\
K(f) &= \left(1+\frac{1}{K_1 f}\left(1+\frac{1}{K_2 f}\right)\right)^2, \textit{and} \\
K(f) &= \left(1+\frac{1}{K_1 f}\left(1+\frac{1}{K_2 f}\right)\left(1+\frac{1}{K_3 f}\right)\right)^2.
\end{aligned}
$$

The coefficients K_i must be input in the array CK.

$$M(f) = (1+M_1 f\,(1+M_2 f)\,(1+M_3 f))^2$$

where any or all of the M_i can be 0. These M_i are stored in the CM array.

This accommodates the most commonly encountered types of random noise found in oscillators, amplifiers, frequency multipliers, frequency synthesizers and general signal processing equipment. In addition the upper cutoff frequency of the integral can be treated as infinitely sharp by setting f_h equal to the equivalent noise bandwidth of the time domain configuration simulated by the calculations, or it can be treated more realistically using the appropriate first-, second-, or third-order low-pass filter function. This option is implemented by choosing the appropriate constants for M_1, M_2, and M_3, and setting f_h to a sufficiently large value that the termination of the integration at this value does not significantly bias the results.

Servo analysis, such as the examination of the effect of various servo parameters on the locking of an oscillator to a frequency reference, is facilitated by choosing the appropriate form of K(f). The noise type assumed for the reference source is white frequency modulation. An example of this is given in the Appendix. In some special servo cases, it may be necessary to divide $S_y(f)$ into two segments and run them as separate cases , or you can enter your own form for $S_y(f)$.

The rest of the parameters are described in the Input Parameter section, below.

2 Running the Program

The SIGINT package of routines is set up for interactive use. It is machine independent with the exceptions noted in the Portability section below. The package will first display the values of the parameters that define the default case. It then presents a menu which asks whether you want to run that case, be prompted to input an entire new set of parameters, or modify directly the current values of some parameters. This will be referred to later as the "input menu".

2.1 Input Modes

In either of the modes described below, values may be entered in either fixed- or floating-point format. If floating-point is used, the decimal point is necessary; for example, you can enter 2.e-3 but not 2e-3. In some cases, when an integer is being entered, the decimal point must be omitted. In all cases, the program recovers from incorrect input, outputs a message attempting to diagnose the problem, and allows re-entry of the value.

2.1.1 The "prompting" input mode

If you elect to be prompted for the values of the variables, the program will lead you through a series of questions and answers by which a new case will be defined. An advantage of this mode is that it is not necessary to know the names of the variables which the program uses.

2.1.2 The "direct" input mode

If, instead, the "direct" mode of input is chosen, values are changed by typing one or more lines containing variable names followed by "=" and the new values of the variable. Different entries can be separated by a comma or blanks. The last line must be terminated by a "$" or a ";". For example, you might type the following lines:

```
NRANGE=2, NLOW = 10
SELSY=1   CK=1.,2.,3. $
```

This mode will usually be faster than the "prompting" mode.

Notice that if you want to get out of this mode without changing anything, you can simply type a "$" or a ";" .

When you finish with either mode of input, the current set of parameters will be displayed and you will be returned to the input menu. At this point you can either run the case or further modify the parameters by again entering one of the input modes. You can continue to revise and examine the parameters until you are satisfied with them and then run the case.

2.2 Output Modes

When the integration is completed, you will be presented with another menu which will be called the "output menu". This menu asks whether to print results, plot them, compute another integral, or quit.

As the program computes an integral, it saves the defining parameters and the results in internal data structures. The results are in the form of a table of values of τ ($\tau = n\tau_0$) and corresponding values of the square root of the integral(variance). So each time you come to the output menu, the results of all of the cases you have done during the run are available to be printed and/or plotted.

2.2.1 Printing

On AT compatible computers you can either print to the screen or to your printer (see Portability section for behavior on other systems).

2.2.2 Plotting

Plotting will work only on AT compatible computers. On each frame it is possible to plot either one or several cases. A hard copy of the plot can be obtained if you have a memory-resident utility that allows you to use the print screen key to transfer the graphics screen image to your printer. The plot package will plot directly to certain printers. Examples of the plots are found in Figures 1-7. Additional details can be

obtained by contacting the Applied and Computational Mathematics Division, NIST, 325 Broadway, Boulder, Colorado 80303-3328. Attention: Abbie O'Gallagher.

2.2.3 Quitting and Saving Results

When you choose to quit, you will be asked if you want to save the results. If you say yes, the defining parameters, along with the results of each of the cases you have done during that run, will be saved in a disk file called SAVOUT on the default disk and directory.

2.3 Input Parameters

The following is a list of the input variables that must be set in order to compute an integral. The integral can be computed for a single value of τ ($\tau = n\tau_0$), or for a sequence of values of τ. In the latter case, the integral can be plotted as a function of τ.

INTGRAL: Integer which selects the integral to be computed. INTGRAL = 1 for the σ integral; INTGRAL = 2 for the modσ integral. The default value is INTGRAL = 1.

NRANGE: Integer which determines the range of values of the parameter n for which the integral is computed. Set NRANGE = 1 to compute the integral for a single value of n (n = NLOW). Set NRANGE = 2 to compute the integral for the sequence of values obtained by starting with NLOW and doubling the current value to obtain the next one until reaching the last such value not exceeding NHIGH. Thus if NRANGE = 2, NLOW = 1, and NHIGH = 20, then the integral will be computed for n=(1, 2, 4, 8, 16). For NRANGE = 3, the integral will be evaluated by decades with five points per decade. Thus if NRANGE = 3, NLOW = 1, NHIGH = 100, then the evaluation will be done for n = (1, 2, 3, 5, 7, 10, 20, 30, 50, 70, 100). The default value is NRANGE = 3.

NLOW: The lower limit of the range of values of n for which the integral is computed (see the discussion of NRANGE). The default value is NLOW = 1.

NHIGH: The upper limit of the range of values of n for which the integral is computed. The default value is NHIGH = 1000.

FH: The upper limit of the integral, f_h. The default value is FH = 3.

SELSY: Integer to select the function $S_y(f)$. If SELSY = 1, you must supply a double-precision function named SYF(f), with double-precision argument f. (You must replace the empty FORTRAN function SYF(f) that is provided.) If SELSY = 2, then you must select one of four built-in functions by giving values of the input parameter SELK, and the arrays C, CK, and CM as given in the discussion of $S_y(f)$ above. The default value of SELSY is SELSY = 2.

SELK: Integer parameter which selects the function K(f). See discussion above. The default value is SELK = 3.

C: A double-precision array of dimension C(6) containing the coefficients of the function $S_y(f)$. The default values are C = 2.e-24, 0, 0, 0, 0, 0.

CK: A double-precision array of dimension CK(3) containing the coefficients of the function K(f). The default values are CK = 10, 40, 100.

CM: A double-precision array of dimension CM(3) containing the coefficients of the function M(f). The default values are CM = 0 ,0, 0.

TAU0: A double-precision scalar parameter of the integral. The default value is TAU0 = 1.

2.4 Internal Constants

It is not necessary to be aware of the constants described in this section unless you want to alter the behavior of the code. In that case it may be necessary to change the values to which these constants are set. This may be done by editing the FORTRAN to change the assignment statements that set these values. Recompiling will also be necessary. The pertinent assignment statements can be found at the beginning of the main program.

EPSREL: The relative error tolerance for QUADPACK routines. It is set at $2 * 10^{-3}$.

EPSTRN: The truncation error tolerance used by the MDSIGY routine to determine, for a given value of the variable of integration, whether the estimate of the integral over the remainder of the interval is small enough to indicate that the remainder can be ignored. See discussion of the numerical method, below. EPSTRN is set at $2 * 10^{-2}$.

NCYCLE: The number of cycles of the numerator of $S_y(f)$ on each side of a singularity which are evaluated by the non-oscillatory quadrature routine (see section 3). NCYCLE is set to 8.

IDB: Switch to control debug printout from the routines SIGMAY and MDSIGY. It is set to 0. Set it to 1 to get debug printing.

3 The Numerical Method

The sigma integral,

$$\sigma_y(n\tau_0) = \left[2\int_0^{f_h} S_y(f)\frac{\sin^4(\pi f n\tau_0)}{(\pi f n\tau_0)^2}df\right]^{\frac{1}{2}},$$

can be treated as an oscillatory integral except in the neighborhood of the origin. Therefore, the integral is computed by using one approximation in the neighborhood of the origin and a second method away from the origin; that is,

$$\sigma_y = [2(I_1 + I_2)]^{\frac{1}{2}},$$

where

$$\begin{aligned} I_1 &= \int_0^{\alpha} F_1(f)df, \\ I_2 &= \int_{\alpha}^{f_h} F_2(f)\sin^4(\pi f n\tau_0)df, \\ F_1(f) &= S_y(f)\frac{\sin^4(\pi f n\tau_0)}{(\pi f n\tau_0)^2}, \text{and} \\ F_2(f) &= \frac{S_y(f)}{(\pi f n\tau_0)^2}, \end{aligned}$$

with α chosen so that the first integral is taken over NCYCLE/2 cycles of the sine function, provided $\tau = n\tau_0$ is large enough. Otherwise it is taken over 2/3 of the interval f_h. That is, $\alpha = minimum(2f_h/3, \text{NCYCLE}/\tau)$. If the function F_1 grows more slowly than f^{-3} at $f = 0$, then the first integral exists. We assume that $S_y(f)$ grows no more rapidly than f^{-2}, and thus the integrand in this first integral has a removable singularity and can be computed using a general purpose quadrature routine from the subroutine library. We use the DQAGE routine from the QUADPACK [7] library.

The second integral is oscillatory. It can be transformed into a canonical form by use of the trigonometric substitution,

$$\sin^4(\pi f\tau) \;=\; \frac{3}{8} - \frac{1}{2}\cos(2\pi f\tau) + \frac{1}{8}\cos(4\pi f\tau),$$

and can therefore be written as the sum of three integrals,

$$I_2 \;=\; I_{21} + I_{22} + I_{23}.$$

Since α is chosen large enough to avoid the singularity at the origin, the first of these three integrals can be computed with the standard quadrature routine DQAGE. The remaining two integrals are oscillatory and are therefore computed using the DQAWOE routine, also from QUADPACK. DQAWOE is designed for integrals of the form $\int f(x)\sin(\omega x)dx$ or $\int f(x)\cos(\omega x)dx$,where ω can be very large. These sigma integrals may have values of $n\tau_0$ which easily exceed 10^4 (here $n\tau_0$ corresponds to ω).

In all cases we ask that the QUADPACK routines compute the integrals with a relative error less than $2 * 10^{-3}$. This may be wasteful, since one of these integrals may dominate the remainder, however we could not be certain of the dominance in all cases.

Extensive tests of the numerical calculation for $\sigma_y(\tau)$ have been performed for all the common noise types, $C_1 - C_5$ [1,2,3]. The results agree with the asymptotic form ($2\pi f_h\tau \gg 1$) to better than 11% as long as $2\pi f_h\tau$ is larger than 6.3 and 0.1% for $2\pi f_h\tau \geq 62.8$. We believe that the numerical technique yields more accurate results since the analytical calcuations are valid only for $2\pi f_h\tau \gg 1$. The asymptotic forms are considered in more detail in Figures 1-7 and Tables I and II.

The mod sigma integral,

$$\mathrm{mod}\sigma_y(n\tau_0) = \left[\frac{2}{n^4\pi^2\tau_0^2}\left\{n\int_0^{f_h}\frac{S_y(f)}{f^2}\sin^4(\pi f n\tau_0)\,df + 2\int_0^{f_h}\sum_{k=1}^{n-1}(n-k)\frac{S_y(f)}{f^2}\cos(2\pi f k\tau_0)\sin^4(\pi f n\tau_0)\,df\right\}\right]^{\frac{1}{2}},$$

is much more difficult to approximate because the sum over k includes many terms when n is large. The sum over k is eliminated by using the following identity,

$$\sum_{k=1}^{n-1}(n-k)\cos\theta k = -\frac{n}{2}+\frac{1}{2}\frac{\sin^2\frac{\theta n}{2}}{\sin^2\frac{\theta}{2}}.$$

This identity is derived by summing a geometric series composed of complex exponetials. Since the real part of a complex exponential is the cosine term, this series reduces to the expression on the left, above. This can be summed explicitly, yielding the final formula. Application of this identity to the original integral leads to the following form of the integral:

$$\mathrm{mod}\sigma_y(n\tau_0) = \left[\frac{2}{n^4\pi^2\tau_0^2}\int_0^{f_h}\frac{S_y(f)\sin^6(\pi\tau_0 nf)}{f^2\sin^2(\pi\tau_0 f)}df\right]^{\frac{1}{2}}.$$

With this simplification $\mathrm{mod}\sigma_y(n\tau_0)$ can be broken up and evaluated in a way that is analogous to that described above for the $\sigma_y(n\tau_0)$ integral. However $\mathrm{mod}\sigma_y(n\tau_0)$ presents the additional problem that it has a singularity for each integral value of $\tau_0 f$ instead of just one singularity at $f = 0$. It is therefore necessary to break up the interval of integration into $f_h * \tau_0$ subintervals and sum the values of the integral over each of these smaller intervals. If f_h is large, this is a large number of subintervals. But in many cases, the later subintervals contribute negligibly to the value of the integral. For this reason an estimate of the error produced by neglecting these contributions is made every few subintervals. If the ratio of this error to the computed value of the integral up to that point is small (less than EPSTRN), then no further calculation is done and the value returned by the program as $\mathrm{mod}\sigma_y(n\tau_0)$ is the square root of the sum of the integrations done over the subintervals up to that point.

Extensive tests of the numerical calcuation for $\mathrm{mod}\sigma_y(\tau)$ have been carried out for the common types of noise ($C_1 - C_5$) and compared to the results of [3] , obtained

analytically and the results of [4] obtained by computer simulated noise. This has been done by comparing R(n) the ratio of mod$\sigma_y^2(\tau)$ to $\sigma_y^2(\tau)$ as a function of the number of samples averaged together for mod$\sigma_y(\tau)$, obtained by the three methods. The results obtained using SIGINT for R(n) are shown in figure 1 and table 1. This agrees exactly with the results of both [3] and [4] for $\alpha = -2$, $\alpha = 0$ and $\alpha = 2$ (ignoring the obvious typographical errors). The results for $\alpha = 1$ agree for $2\pi f_h \tau_0 = 10^4$ - the only case addressed by [3]. Also shown are our results for $\alpha = 1$ and $2\pi f_h \tau_0 = 3$, 10, 100. Our results differ considerably from [3] for $\alpha = -1$, but agree with the asymptotic value of [4]. Since we have obtained excellent agreement for so many other cases using the same numerical code, we believe that the results presented in figure 1 and table 1 are more accurate than the earlier work.

4 Portability

The SIGINT code is standard FORTRAN 77 and is portable with the following exceptions:

- The graphics library used is David Kahaner's[1] GRAPH.LIB, which runs only on PC compatible computers with one of the following display adapters:
  - IBM Color Graphics Adapter[2]
  - IBM Enhanced Graphics Adapter
  - IBM Professional Graphics Adapter in CGA emulation mode
  - IBM PS/2 Video Gate Array in EGA emulation mode
  - Hercules
  - Video-7

  When moving the code to another system, it will be necessary either to "comment out" the calls to the plot routines and run the code without graphic output

[1]Scientific Computing Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

[2]Identifacation of some commercial materials has been necessary in this report. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials are necessarily the best available for the purpose.

or to change those graphics calls, all of which are in SUBROUTINE PLOT, to call an available graphics library.

- On any system, numerical results can be printed on the screen but if the option of using the printer is chosen on a system other than an AT compatible, the results will instead be written to a file called "PRN". ("PRN" is the name DOS uses for the printer.) Results will not accumulate in this file. Only the last printing will be there when the program terminates.
- The code is written in double precision, which may be unnecessary on machines with larger word sizes.
- On UNIX systems, carriage control characters may appear on the screen instead of affecting the output as intended.
- The VAX reports underflow conditions but terminates normally and the results are unaffected.

The entire code has been run extensively on AT compatible computers using RM/FORTRAN. In addition, except for the graphic output, it has been tested on a VAX 11/785, SUN 3/180, MASSCOMP MC5500-PEP and Cyber 840 and 855. It is expected that it will run on most other systems. If problems are encountered in moving it to another system, an attempt will be made, subject to the availability of resources, to overcome them. For such assistance, please contact the Applied and Computational Mathematics Division, NIST, 325 Broadway, Boulder, Colorado 80303-3328. Attention: Abbie O'Gallagher.

References

[1] James A. Barnes, Andrew R. Chi, Leonard S. Cutler, Daniel J. Healey, David B. Leeson, Thomas E. McGunigal, James A. Mullen,Jr., Warren L. Smith, Richard L. Sydnor, Robert F. C. Vessot, and Gernot M. R. Winkler, Characterization of Frequency Stability, IEEE Transactions on Instrumentation and Measurement, 1971, IM-20, 105-120.

[2] David Allan, Helmut Hellwig, Peter Kartaschoff, Jacques Vanier, John Vig, Gernot M. R. Winkler, and Nicholas Yannoni, Standard Terminology for Fundamental Frequency and Time Metrology, Proceedings of the 42d Annual Frequency Control Symposium , 419-425, 1988.

[3] Paul LeSage and Théophane Ayi, Characterization of Frequency Stability: Analysis of the Modified Allan Variance and Properties of Its Estimate, IEEE Transactions on Instrumentation and Measurement, 1984, IM-33, 332-337.

[4] David W. Allan and James A. Barnes, A Modified "Allan Variance" with Increased Oscillator Characterization Ability, Proceedings of the 35th Annual Frequency Control Symposium, 470-475, 1981.

[5] V. F. Kroupa and L. Sojdr, Modified Allan Variance: Transfer Functions and Practical Applications, CPEM Digest (1984).

[6] P. Tremblay, Etude de l'Effet du Filliage du Sigmal sur la Caracterisation de la Stabilite de Frequence, These Universite, Laval, Quebec (1985).

[7] R. Piessens, E. deDoncker-Kapenga, C.W. Uberhuber, and D.K. Kahaner, QUADPACK – A Subroutine Package for Automatic Integration, Springer-Verlag, Berlin, 1983.

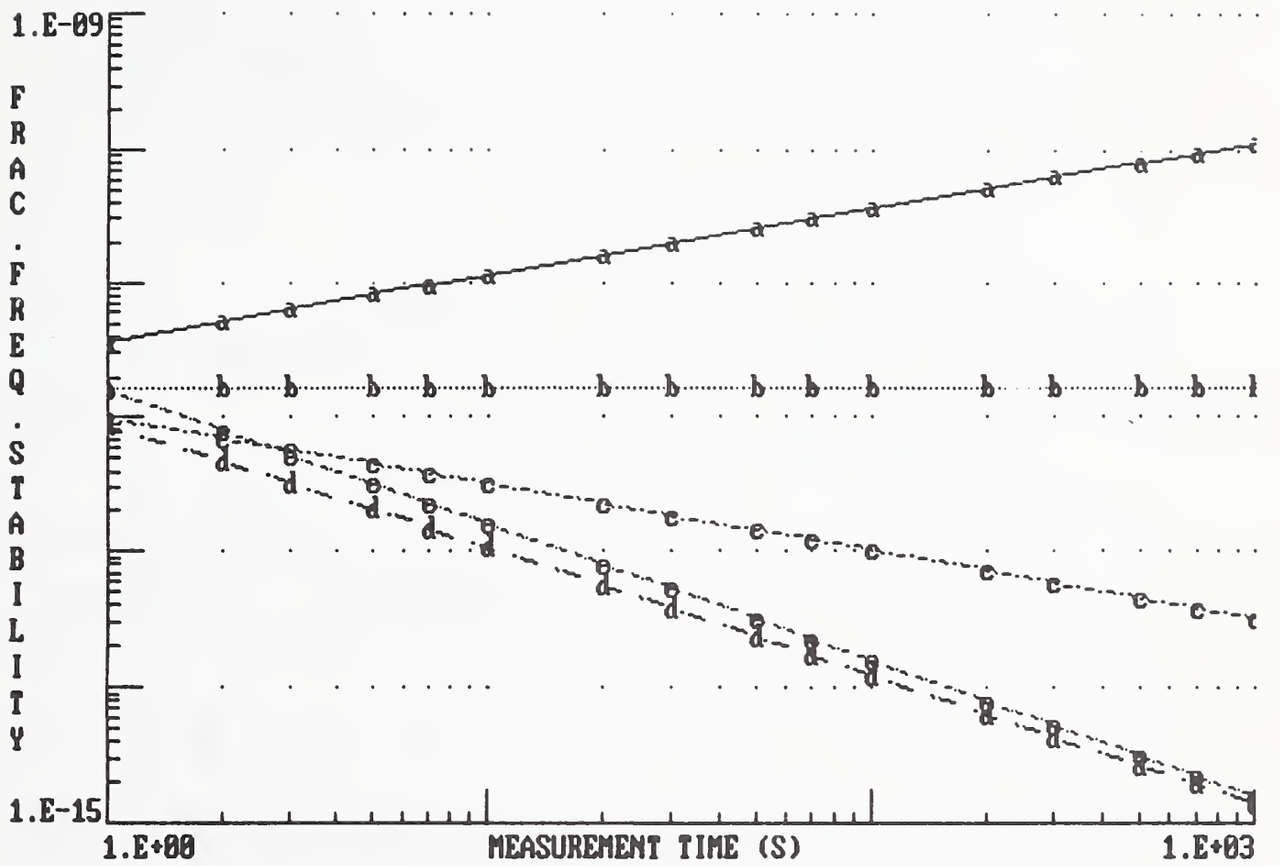

Figure 1. $\sigma_y(\tau)$ versus τ for the five common power-law noise types in the limit that $2\pi f_h$ is large compared to 1 and an infinitely sharp filter is used. Curve a is for random-walk frequency modulation, $S_y(f) = h_{-2} f^{-2}$. Curve b is for flicker frequency modulation, $S_y(f) = h_{-1} f^{-1}$. Curve c is for white frequency modulation, $S_y(f) = h_0$. Curve d is for flicker phase modulation, $S_y(f) = h_1 f$. Curve e is for white phase modulation, $S_y(f) = h_2 f^2$.

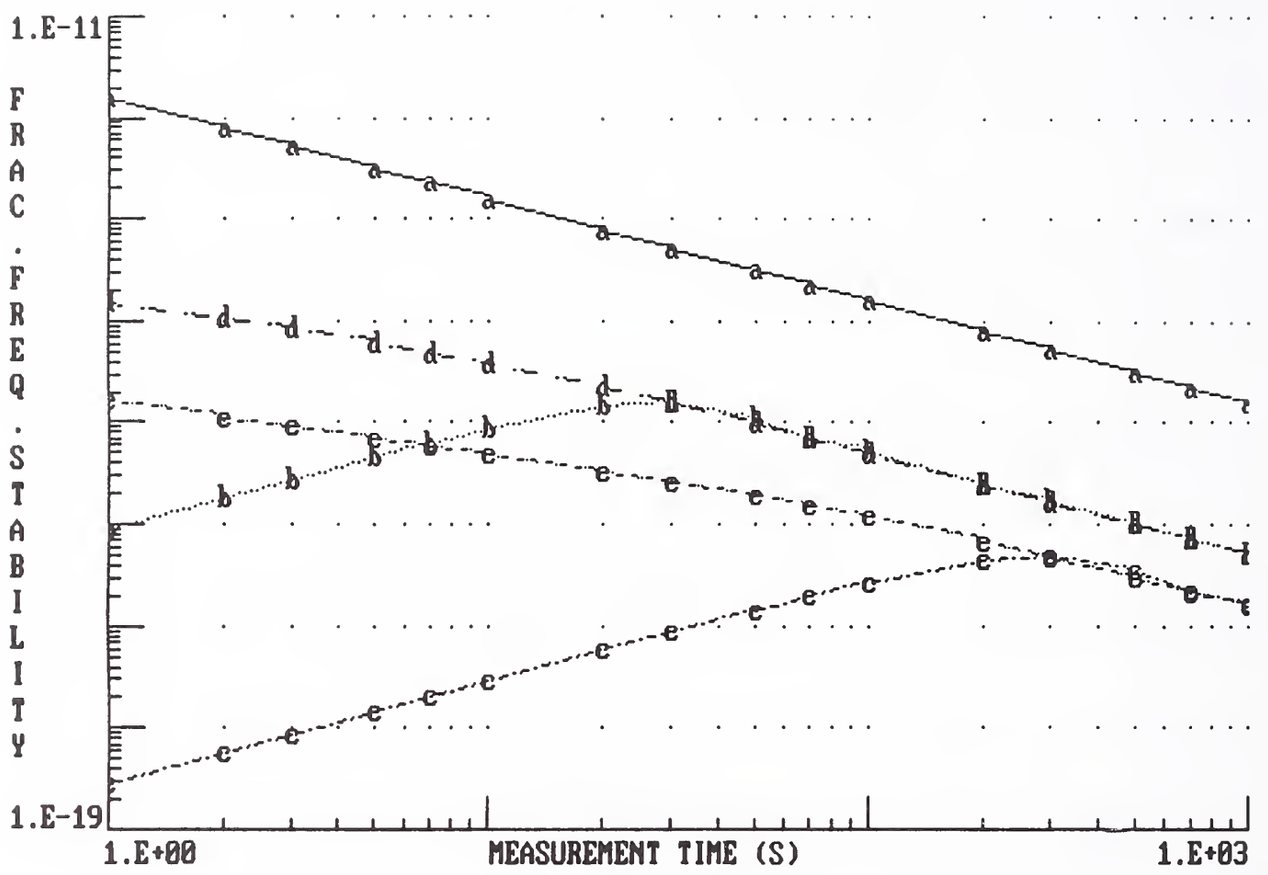

Figure 2. $\sigma_y(\tau)$ for white phase modulation ($\alpha = 2$) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b, and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively.

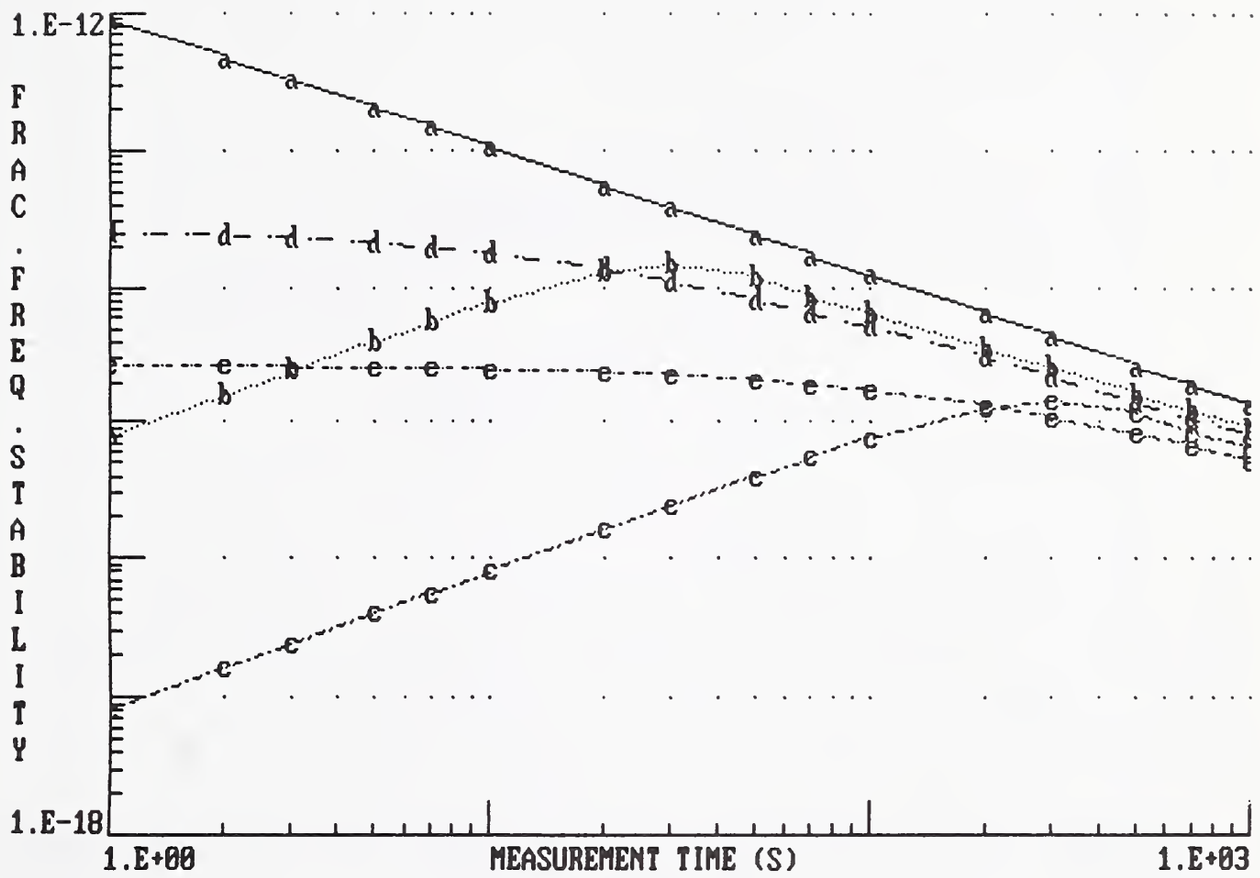

Figure 3. $\sigma_y(\tau)$ for flicker phase frequency modulation (α = 1) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b, and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively.

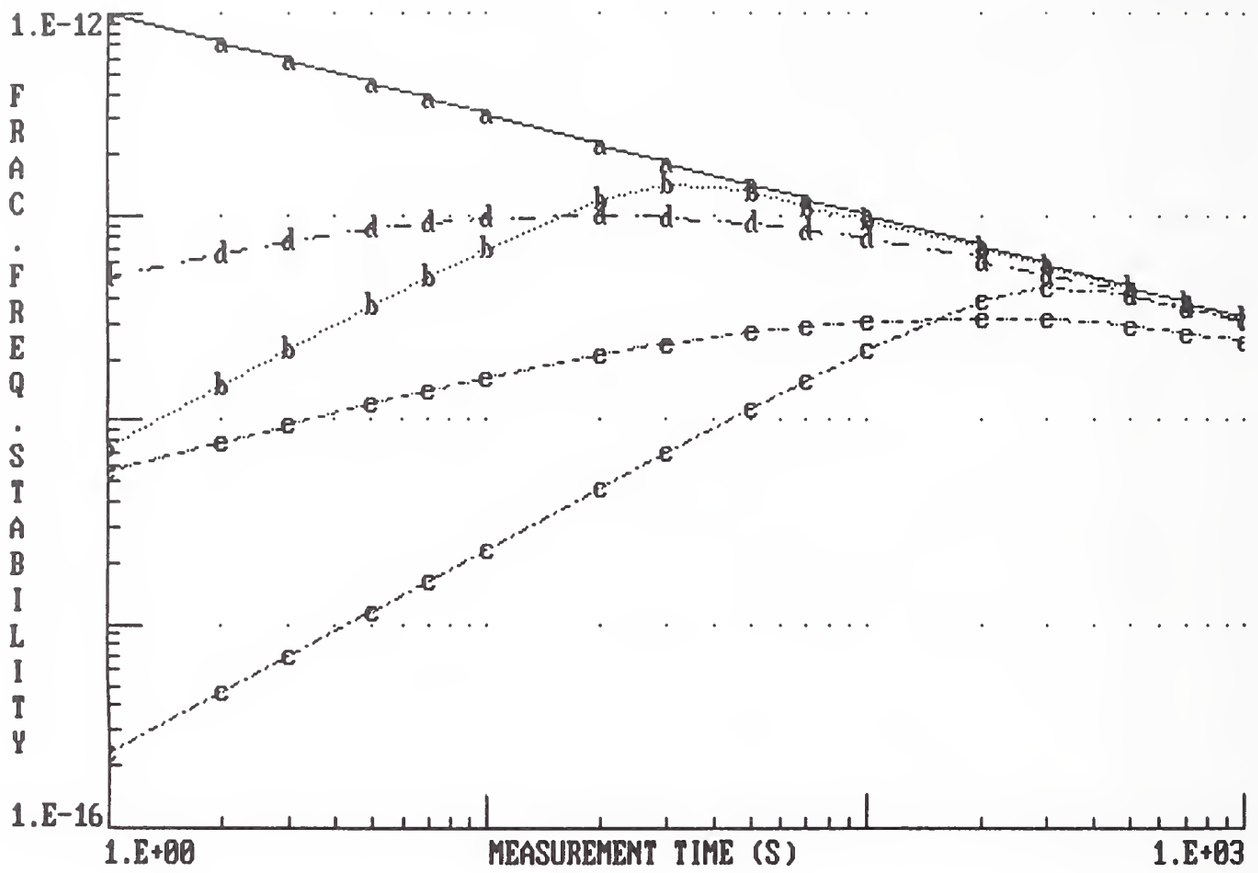

Figure 4. $\sigma_y(\tau)$ for white frequency modulation ($\alpha = 0$) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b, and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively.

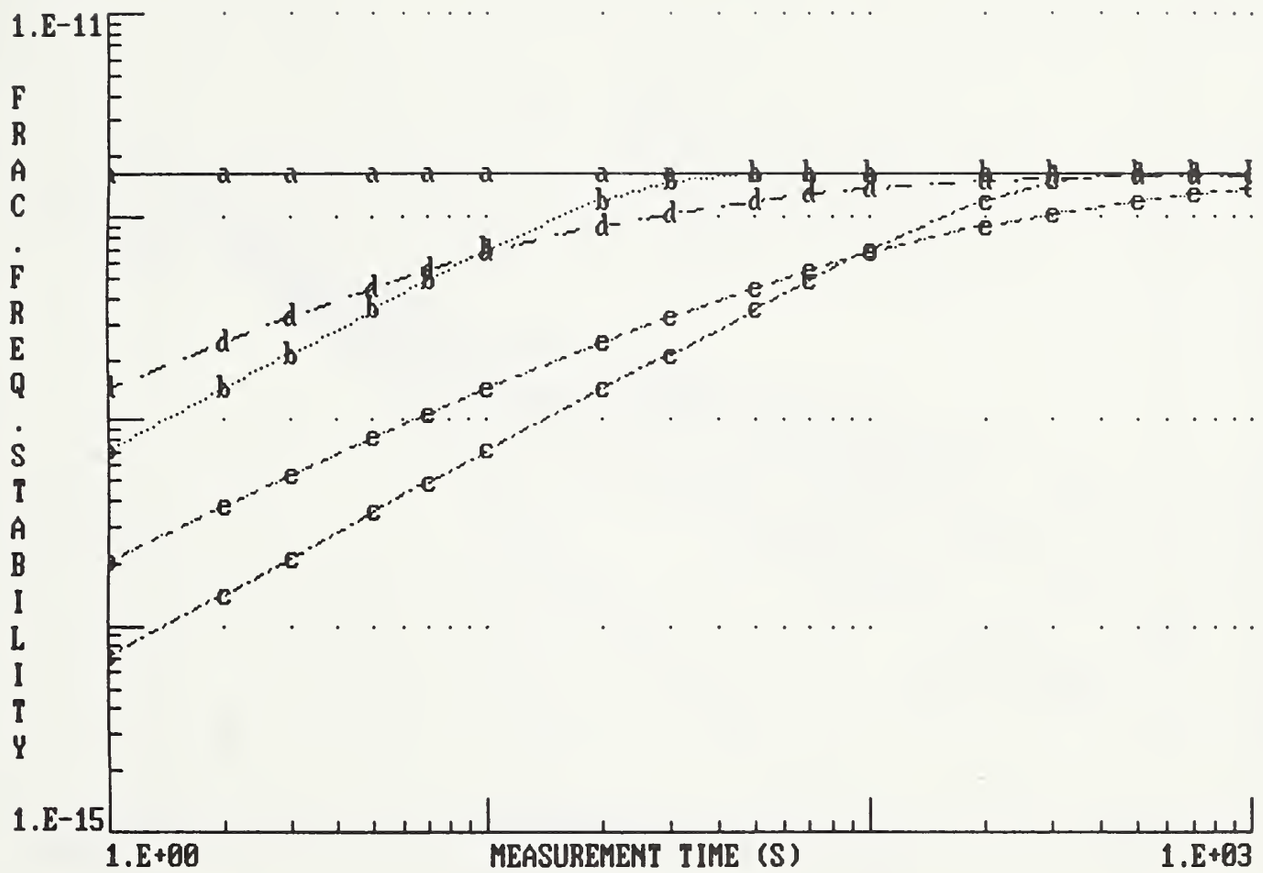

Figure 5. $\sigma_y(\tau)$ for flicker frequency modulation (α = -1) as a function of measurement time, τ, and measurement bandwidth, f_h. Curves a, b, and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively.

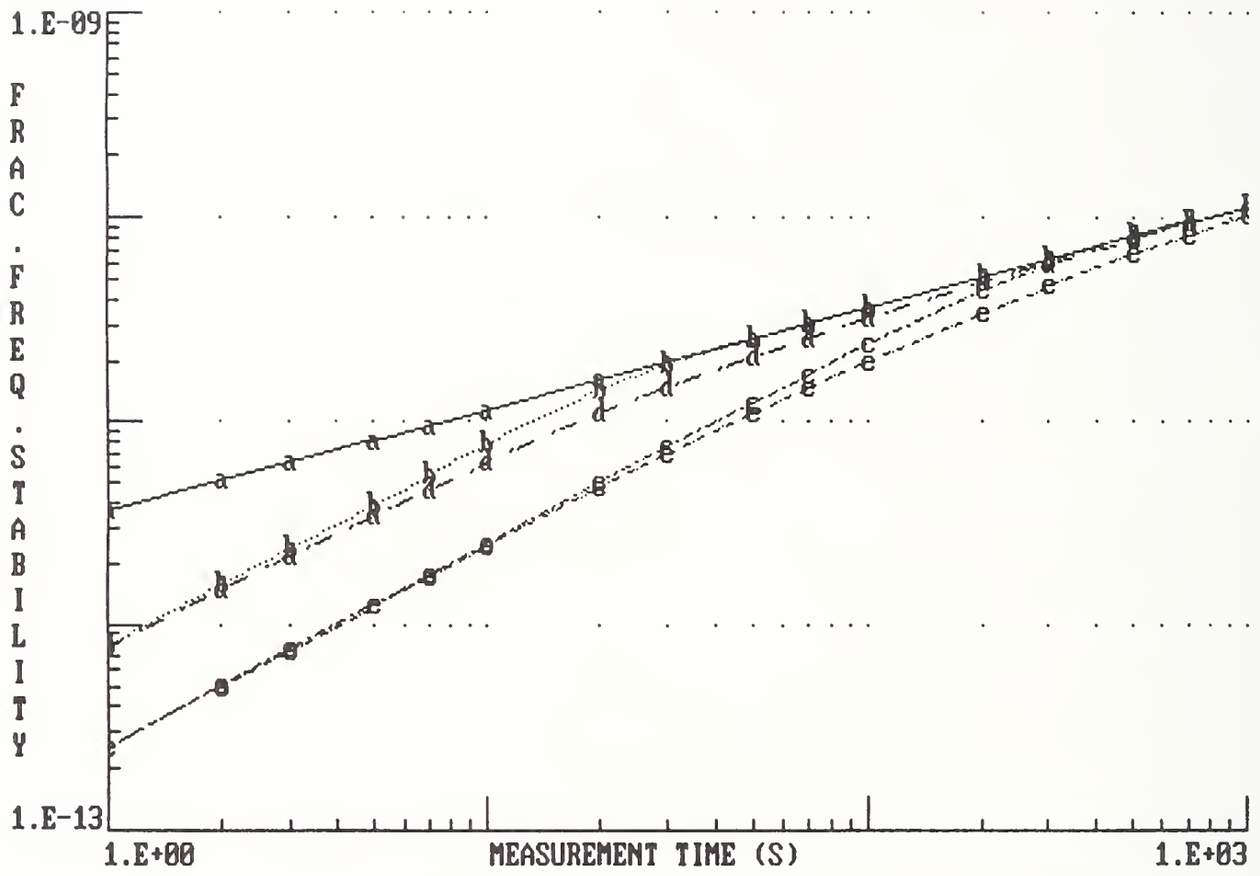

Figure 6. $\sigma_y(\tau)$ for random walk frequency modulation (α = -2) as a function of measurement time and measurement bandwidth, f_h, for π. Curves a, b, and c have an infinitely sharp filter with width, f_h = 16 Hz, f_h = 0.016 Hz, f_h = 0.0016 Hz respectively. Curves d and e have a single pole filter width, f_h = 0.016 Hz and f_h = 0.0016 respectively.

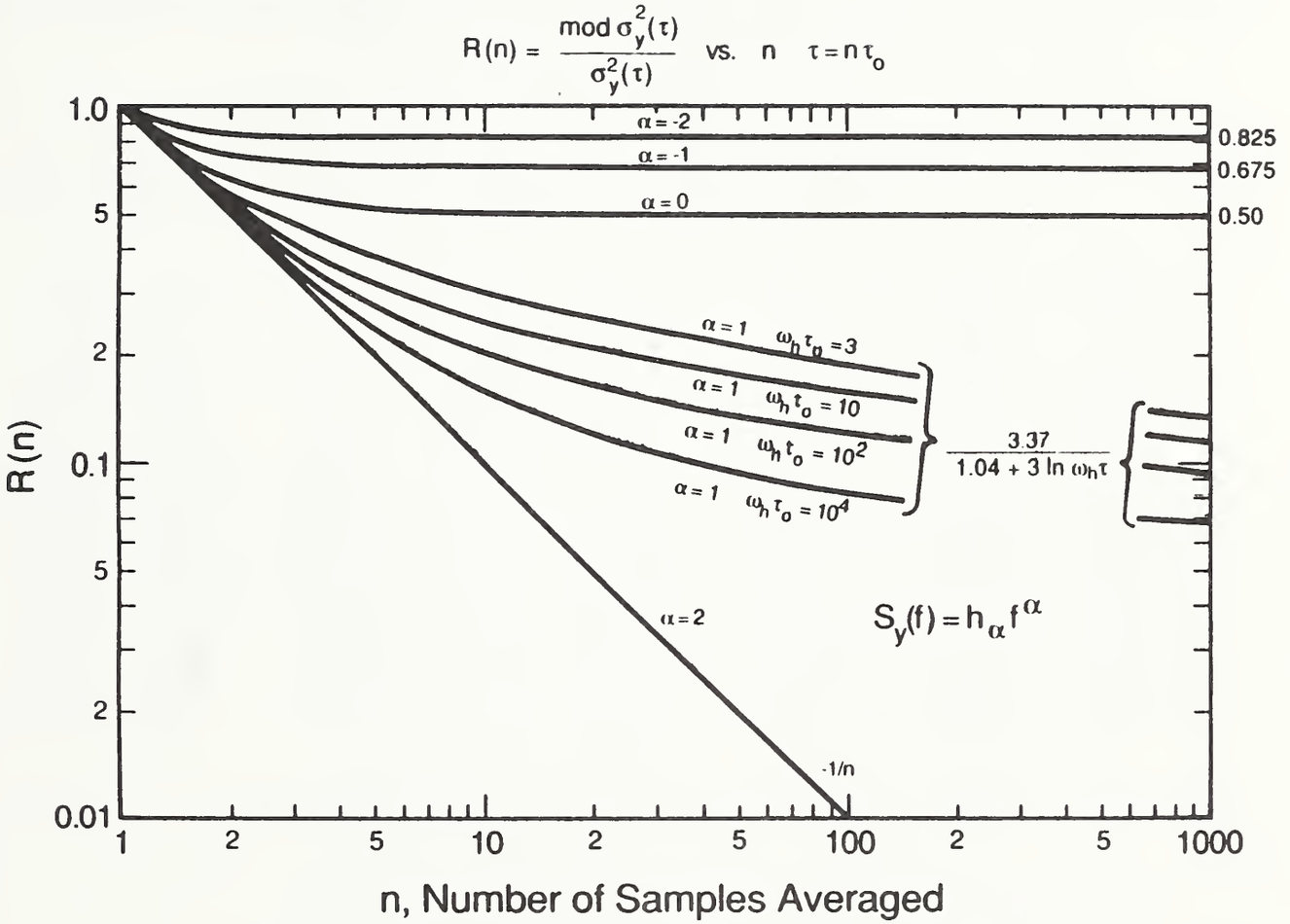

Figure 7. Ratio of mod$\sigma_y(\tau)^2$ to $\sigma_y(\tau)^2$ as a function of n, the number of points averaged to obtain Mod$\sigma_y(\tau)$. The measurement time $\tau = n\tau_0$, where τ_0 is the minimum data interval.

TABLE I. Ratio of $\mathrm{mod}\sigma_y^2(\tau)$ to $\sigma_y^2(\tau)$ vs n, for common power-law noise types $S_y(f) = h_\alpha f^\alpha$. n is the number of time or phase samples averaged to obtain $\mathrm{mod}\sigma_y^2(\tau = n\tau_0)$ where τ_0 is the minimum sample time, and ω_h is 2π times the measurement bandwidth f_h.

$$R(n) = \frac{\mathrm{mod}\,\sigma_y^2(\tau)}{\sigma_y^2(\tau)} \quad \text{vs.} \quad n \quad \tau = n\tau_0$$

n	$\alpha = -2$	$\alpha = -1$	$\alpha = 0$	$\alpha = +1$ $\omega_h\tau_0 = 3$	$\omega_h\tau_0 = 10$	$\omega_h\tau_0 = 100$	$\omega_h\tau_0 = 10^4$	$\alpha = +2$
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	0.859	0.738	0.616	0.568	0.543	0.525	0.504	0.500
3	0.840	0.701	0.551	0.481	0.418	0.384	0.355	0.330
4	0.831	0.681	0.530	0.405	0.359	0.317	0.284	0.250
5	0.830	0.684	0.517	0.386	0.324	0.279	0.241	0.200
6	0.828	0.681	0.514	0.349	0.301	0.251	0.214	0.167
7	0.827	0.679	0.507	0.343	0.283	0.235	0.195	0.143
8	0.827	0.678	0.506	0.319	0.271	0.219	0.180	0.125
10	0.826	0.677	0.504	0.299	0.253	0.203	0.160	0.100
14	0.826	0.675	0.502	0.274	0.230	0.179	0.137	0.0714
20	0.825	0.675	0.501	0.253	0.210	0.163	0.119	0.0500
30	0.825	0.675	0.500	0.233	0.194	0.148	0.106	0.0333
50	0.825	0.675	0.500	0.210	0.176	0.134	0.0938	0.0200
100	0.825	0.675	0.500	0.186	0.159	0.121	0.0837	0.0100
Limit	0.825	0.675	0.500	$\longleftarrow \left(\frac{3.37}{1.04 + 3\ \ln\omega_h\tau}\right) \longrightarrow$				1/n

TABLE II. Asymptotic forms of $\sigma_y^2(\tau)$ for various power-law noise types and two filter types. Note: $\omega_h/2\pi = f_h$ is the measurement system bandwidth--often called the high-frequency cutoff. ln ≡ loge.

Name of Noise	α	$S_y(f)$	$\sigma_y^2(\tau)$			
			$\omega_h\tau>>1$ Infinite Sharp Filter	$\omega_h\tau>>1$ Single Pole Filter	$\omega_h\tau<<1$ Infinite Sharp Filter	$\omega_h\tau<<1$ Single Pole Filter
White Phase	2	$h_2 f^2$	$\frac{3f_h h_2}{(2\pi)^2\tau^2}$	$\frac{3f_h h_2}{(2\pi)^2\tau^2}$	$2/5\pi^2 f_h{}^5\tau^2 h_2$	$\frac{f_h{}^2 h_2}{2\tau}$
Flicker Phase	1	$h_1 f$	$\frac{(1.038 + 3\ln(\omega_h\tau))h_1}{(2\pi)^2\tau^2}$	$\frac{(3\ln(\omega_h\tau))h_1}{(2\pi)^2\tau^2}$	$\frac{1}{4}\pi^2 f_h{}^4\tau^2 h_1$	$2f_h{}^2(\ln(2))h_1$
White Frequency	0	h_o	$\frac{h_o}{2\tau}$	$\frac{h_o}{2\tau}$	$2/3\pi^2 f_h{}^3\tau^2 h_0$	$2/3\pi^2 f_h{}^2\tau h_0$
Flicker Frequency	-1	$h_{-1} f^{-1}$	$2(\ln(2))h_{-1}$	$2(\ln(2))h_{-1}$	$\pi^2 f_h{}^2\tau^2 h_{-1}$	$8\pi^2 f_h{}^2\tau^2 h_{-1}$
Random-Walk Frequency	-2	$h_{-2} f^{-2}$	$\frac{2\pi^2\tau h_{-2}}{3}$	$\frac{2\pi^2\tau h_{-2}}{3}$	$2\pi^2 f_h\tau^2 h_{-2}$	$2\pi^2 f_h\tau^2 h_{-2}$

Appendix

Stability of Frequency Locked Loops
Fred L. Walls
National Institute of Standards and Technology
Boulder, Colorado 80303

1. Introduction

Passive frequency standards are characterized by the use of a reference resonance to stabilize the frequency of an external probe oscillator. A common configuration is shown in Fig. 1 [1-6]. The probe oscillator generally has phase modulation imposed on the carrier in order to interrogate the resonance with a minimum of offset. The resulting amplitude modulation is demodulated to yield an error curve that is essentially the derivative of the resonance. Although Fig. 1 shows a transmission system, similar schemes are sometimes possible in reflection [5]. The error signal from the demodulator is used to steer the probe signal toward the center of the resonance line [1-9]. For analysis times longer than one period of the modulation cycle, and under the condition that the probe oscillator wanders less than the half width of the error curve in the loop attack time, we can treat this curve as approximately static[1-4]. Near line center the loop error voltage V_D at the synchronous detector, is approximately $V_D = k\ (\nu_o° - \nu_R) + V_n$ where k is the slope of the error curve, ν_R is the resonance frequency of the reference, and $\nu_o°$ is the open loop frequency of the probe oscillator and V_n is the detector noise. If we now close the loop with gain G(f), it can be shown that the spectral density of fractional frequency fluctuations $S_y(f)$ for the probe source becomes

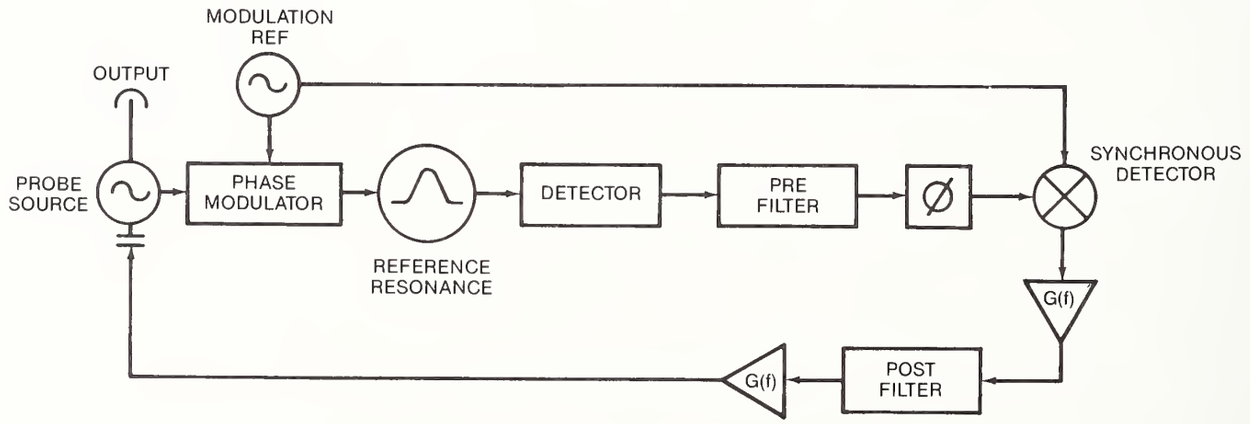

Figure 1. Generalized block diagram of a probe source locked to a reference resonance.

Contribution of the U.S. Government; not subject to copyright.

$$S_y(f) = \left[\frac{G(f)}{G(f)+1}\right]^2 \left[S_y{}^R + \frac{S_y{}^\circ(f)}{G^2(f)}\right] + S_y{}^N(f) \qquad (1)$$

where $Sy^\circ(f)$ is the open-loop spectral density of fractional frequency fluctuations of the probe source, $Sy^R(f)$ is that of the reference, and $Sy^N(f)$ is that of the detector and interrogation noise referred to the demodulator output. Cutler [10] has pointed out that noise in the local oscillator at the 2nd harmonic of the modulation frequency, which is usually ignored, causes a time varying frequency offset that is undistinguishable from reference noise. This sets the lower limit to the interrogation noise and often sets the lower limit of the noise performance of the local oscillator necessary not to degrade the overall performance. The magnitude of the 2nd harmonic noise modulation in radians/s is estimated in appendix B of [1] to be $k_3 = 2\pi\nu_(S_y(\Omega/\pi))^{1/2}$, where $\Omega/(2\pi)$ is the modulation frequency. This leads to an interrogation noise term which is of order [1,4,10] $S_y{}^N(f) = 1/(16\pi)\ S_y{}^\circ(\Omega/\pi)$. For large values of G(f), $S_y(f)$ of the probe source reflects that of the reference plus the added noise of the detection system [1-6].

The primary goal of this paper is to investigate the effect of various realistic forms of G(f) on the spectral density of frequency and fractional-frequency stability. It will be shown that mod $\sigma_y(\tau)$ [11,12] is better suited than the traditional two-sample or Allan Variance $\sigma_y(\tau)$ [8], for evaluating the locked performance when the frequency stability of the reference is much greater than that of the local oscillator [7,8,11,12].

2. The Effect of Different Forms of Servo Gain

Figure 2 shows the effect of locking a probe source with $S_y{}^\circ(f) = 2 \times 10^{-28}/f^2 + 1 \times 10^{-24}/f + 2 \times 10^{-30} f^2$ (which roughly corresponds to that of a low-noise 5 MHz quartz oscillator) to a reference resonance with $S_y{}^R(f)$ of 2×10^{-30} and ${}_{S}y^N(f) = 5.6 \times 10^{-28}$. $S_y{}^N(f)$ is the estimated interrogation noise for a modulation frequency of 47 Hz [1,4,10]. Curves A, B, and C show the effect of using a first-, second-, or third- order loop each having a loop bandwidth of approximately 0.1 Hz or an attack time of 1.6 s[1]. The solid Curves A, B, and C of Fig. 3 show $\sigma_y(\tau)$ calculated for curves A, B, and C from Fig. 2 with a noise bandwidth of 3 Hz. The improvement in stability of the local oscillator due to the servo scales roughly as τ/τ_o where τ_o is the attack time and τ is the measurement time. Considerable insight into the effect of various gain stages on the frequency stability can be obtained by considering

$$\frac{\Delta\nu}{\nu_0} \sim \sqrt{\Delta\phi^2/(2\pi\nu_R\tau)^2} \qquad (2)$$

as a measure of fractional time or frequency stability in the region of measurement times from about 1 to 10^5 s where the effects of low frequency divergence can often be ignored [7,8]. The squared phase deviation of the zero crossings is given approximately by[8]

$$\Delta\phi^2 = \nu^2{}_R \int_{1/2\pi^2\tau}^{f_h} (S_y(f))/f^2 \, df \qquad (3)$$

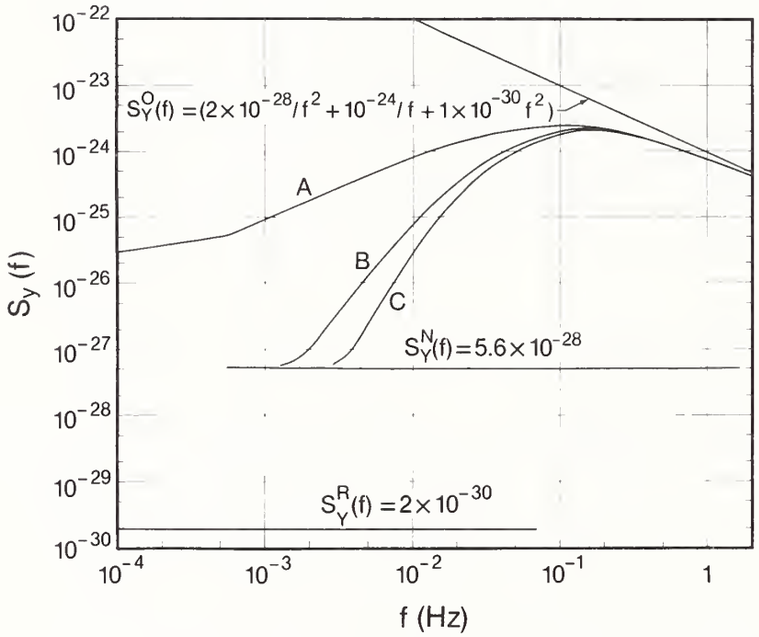

Figure 2. $S_y(f)$ of the probe source locked to a reference resonance. For A G(f) = (1/(10f), for B G(f) = (1/(10f))(1 + 1/(40f)), for C G(f) = 1/(10f)) (1 + 1/(40f))(1 + 1/(160f)).

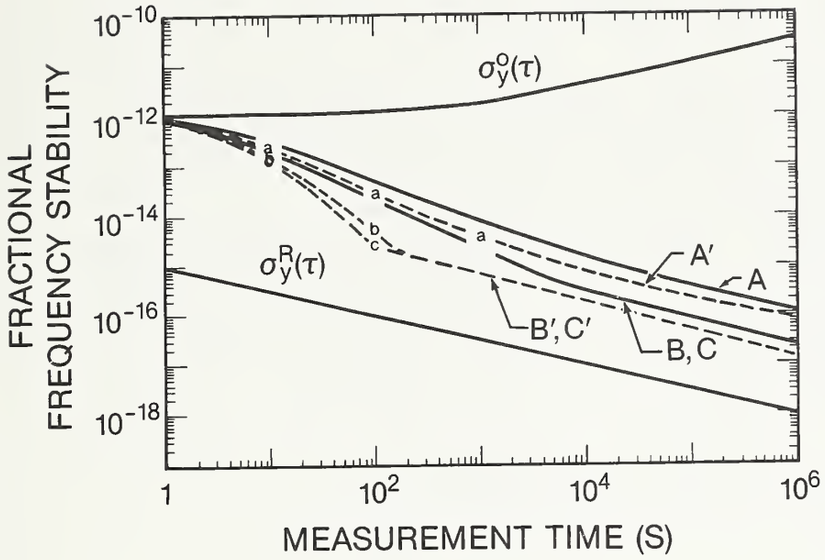

Figure 3. The solid curves show $\sigma_y(\tau)$ and the dashed curves show mod $\sigma_y(\tau)$ versus τ for curves A, B, and C of Fig. 2. The a, b, and c points show estimations of mod $\sigma_y(\tau)$ using Eqs. 2 and 3 with $f_h = f_{ho}$. Also shown is $\sigma_y{}^o(\tau)$ and $\sigma_y{}^r(\tau)$.

for a sample time τ and noise bandwidth f_h. For a noise spectrum which varies as $Sy(f) = K'f^n$ where $n > 2$, the integral is dominated by the high frequency bandwidth even for very long measurement times. The fractional-frequency (or time) stability given by Eq. 2 decreases as τ^{-1} just as does $\sigma_y(\tau)$, due to the increase in measurement time and not to a decrease in the value of $\Delta\phi^2$. The integral in Eq. 3 can be integrated by parts for the various segments of Sy(f). This makes it easier to evaluate and optimize the performance of the overall system than by using a process based on $\sigma_y(\tau)$.

The contribution of the high frequency noise can be greatly reduced by phase averaging the data points[9]. The data at measurement time $\tau = n\tau_o$ (where τ_o is the data interval) is obtained by averaging the n adjacent phase points. This is equivalent to using mod $\sigma_y(\tau)$ to analyze the data[9,11,12]. The standard expression for mod $\sigma_y(\tau)$ contains an enormous number of terms and is quite laborious to compute[11-12]. It [13-15] has been pointed out that mod $\sigma_y(\tau)$ can be reduced to

$$\text{mod } \sigma_y^2(\tau) = \frac{2}{n^4\pi^2\tau_o^2} \int_0^{f_h} \frac{S_y(f)\sin^6(\pi\tau_o nf)}{f^2\sin^2(\pi\tau_o f)}\, df. \qquad (4)$$

which is much more manageable. Nevertheless it still requires numerical calculations to determine which segment of the phase noise dominates the integral. We can estimate mod $\sigma_y(\tau)$ from Eqs. 2 and 3 by using $f_h = f_{ho}/n$, where f_{ho} is the hardware bandwidth of the measurement system and

$\tau = n\tau_o$ is the measurement time. The integral for $\Delta\phi^2$ in Eq. 3 is now substantially reduced for measurement times large enough that f_{ho}/n is less than the bandwidth of the servo system. This is in contrast to the original calculation for Eq. 3 where the integral must always increase with τ. This integral can also be divided into parts and integrated analytically. The fractional frequency stability computed for A, B, and C of Fig. 2 using mod $\sigma_y(\tau)$ are shown as the dashed curves A', B', and C' of Fig. 3. The points labeled a, b, c show the results of estimating mod $\sigma_y(\tau)$ using Eqs. 2 and 3 with $f_h = f_{ho}/n$. The agreement with the dashed curves is very good. The calculations for $\sigma_y(\tau)$ and mod $\sigma_y(\tau)$ are virtually independent of using an upper cutoff frequency for the integration or a simple low pass filter of the same bandwidth. The use of mod $\sigma_y(\tau)$ reduces by a factor of about 100 the time necessary to reach the performance of the reference plus the interrogation noise, which in this case limits the performance for times longer than 100 s.

3. Discussion

The primary utility of these results is the insight into the origin of the major contributions to $\sigma_y(\tau)$ and mod $\sigma_y(\tau)$ as a function of the servo gain and the analysis bandwidth, and the effect of narrowing the bandwidth with longer measurement times. The specific examples illustrate that it is generally necessary to use a second-order loop to lock the probe source to the reference resonance, but that a third-order loop offers little additional improvement when drift in the probe is not serious. In addition, if the reference resonance is substantially more stable than the probe source, it can be very useful to use mod $\sigma_y(\tau)$ to analyze the output of the probe source. We have introduced a simple measure of frequency stability that is easy to use for optimizing the servo and the analysis system. With the techniques introduced here it is possible to realize fractional frequency stabilities and time prediction of the local oscillator which are characteristic of the reference resonance in a way that is orders of magnitude faster than those using more traditional approaches.

4. Acknowledgements

It is a pleasure to acknowledge the many enlightening discussions on this topic with L. S. Cutler, A. DeMarchi, R. E. Drullinger, and D. W. Allan. M. Cline made some of the initial calculations, A. O'Gallagher, J. Gary, and R. Sweet wrote the routines to calculate $\sigma_y(\tau)$ and mod $\sigma_y(\tau)$ from the specified phase noise.

5. References

1. F. L. Walls and S. R. Stein, NBS Tech Note 692, U.S. Government Printing Office, Washington, D.C., SD Cat. # C1346:692
2. C. Audoin and V. Candelier, IEEE Trans. on UFFC, UFFC-34, 573-581 (1987)
3. A. DeMarchi, G. D. Rovera, and A. Premoli, IEEE Trans. on UFFC, UFFC-34, 582-591 (1987)
4. F. W. Walls, IEEE Trans. on UFFC, UFFC-34, 592-597 (1987)
5. S. R. Stein, Proc. of 29th Symposium on Frequency Control, 321-327 (1975)
6. J. Vanier, M. Tetu, and L. G. Berniez, IEEE Trans. on I&M IM-28, 188-193
7. D. W. Allan, Proc. IEEE, 54, 221-230 (1966)
8. J. A. Barnes et al., IEEE Trans. I&M, IM-20, 120 (1971)
9. J. J. Snyder, Appl. Opt., 19, 1223-1225 (1980)
10. L. S. Cutler, Hewlett Packard, Private communication, (Sept. 1988).
11. D. W. Allan and J. A. Barnes, Proc. 35th Annu. Symp. Frequency Control, 470-474 (1981)
12. P. Lesage and T. Ayi, IEEE Trans. I&M, IM-33, 332-336 (1984)
13. J. Gary, National Bureau of Standards, Boulder, CO, private communication.
14. V.P. Kroupa and L. Sojar, CPEM 1984 Digest (Delft, the Netherlands.)
15. P. Tremblay, These Université Laval, Quebec 1985.

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 89-3916

2. PERFORMING ORGANIZATION REPORT NUM

3. PUBLICATION DATE
September 1989

4. TITLE AND SUBTITLE

TIME DOMAIN FREQUENCY STABILITY CALCULATED FROM THE FREQUENCY DOMAIN DESCRIPTION: Use of the SIGINT Software Package to Calculate Time Domain Frequency Stability From the Frequency Domain

5. AUTHOR(S)

F.L. Walls, John Gary, Abbie O'Gallagher, Roland Sweet, and Linda Sweet

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
~~GAITHERSBURG, MD 20899~~ Boulder, Colorado 80303-3328

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

[ ] DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY LITERATURE SURVEY, MENTION IT HERE.)

We describe the use of SIGINT, a new interactive software package developed by the National Institute of Standards and Technology, which facilitates the calculation of time domain frequency stability in terms of the Allan variance, $\sigma_y^2(\tau)$, or mod$\sigma_y^2(\tau)$ as a function of measuring time from frequency domain data. Except for the graphic output, the code is written in standard FORTRAN 77 and runs on AT compatible computers that have a math co-processor. It also runs on many other systems; however, calls to an available graphics library will need to be substituted for those that are included in this version. The program uses either a user defined function for the input noise or default functions that describe the noise types commonly found in oscillators, amplifiers, frequency multipliers, frequency dividers, and general signal processing equipment. These default functions make it simple to analyze the time domain frequency stability as a function of measuring bandwidth using realistic first-, second-, or third- order low-pass filters or the simplified infinitely sharp cutoff parameter f_h. The default functions are also set up to examine the effect of various servo parameters on the performance of a frequency source locked to a frequency reference.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

Allan variance; frequency lock loop analysis; modified Allan variance; phase lock loop analysis; spectral density of frequency stability; time domain frequency stability

13. AVAILABILITY

[X] UNLIMITED

[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
36

15. PRICE

ELECTRONIC FORM

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990—775-286/25138

Jun 13, 2016